

An Adaptive Configuration Method to Multi-Cards in Launch Vehicle Ground Test System

Y.W. CAI, G.F.XU

Equipment Academy, Beijing, China

ABSTRACT: Due to the diversity and complexity caused by multi-cards in launch vehicle ground test system, an adaptive configuration method to multi-cards is studied. This method includes 2 stages and 9 steps. After a unified format conversion, nonlinear genetic algorithm (NGA) is introduced into the adaptive configuration process to ensure the intelligently. The generated global optimal configuration parameters would improve the efficiency of system development.

KEYWORD: Adaptive Configuration; Multi-card; Launch Vehicle; Ground Test System

1 INTRODUCTION

At present, the types and quantity of signal that are tested by launch vehicle ground test system is huge. So the types and quantity of test cards is using more and more[1]. The configuration of the test cards has great influence on the performance and efficiency of test system[2]. The function and performance index of the test systems are different. How to configure the test cards fast and automatically is becoming a research hot.

In this paper, an adaptive configuration method to multi-cards is put forward for the first time. According to the system demand, test cards will be configured adaptively by the method. After a certain experimental comparison, optimal configuration parameters of test cards can be found out.

The rest of the paper is organized in the following sequence. In section 2, various parameters of various

test cards will be introduced in detail. In section 3, the design idea and realization process of the adaptive configuration method will be elaborated in detail. Concluding remarks are summarized in the final section.

2 VARIOUS PARAMETERS OF VARIOUS TEST CARDS

In the ground test system, many test cards are used. They are PXI2208A/D analog signal acquisition card, cPCI7300 digital I/O card, CPCI7434 relay card and NI4130 constant current source card etc. Limited to the length of the paper, only PXI2208A/D analog signal acquisition card is chosen to show the diversity and complexity of the interfaces and parameters [3]. The details are shown in table 1.

Table 1. Interfaces and parameters of PXI2208A/D .

Interface	Description
I16 D2K_Register_Card (U16 CardType, U16 card_num)	Initialization
I16 D2K_AI_CH_Config (U16 CardNumber, U16Channel, U16 AdRange_RefGnd)	Configuration of voltage range
I16 D2K_AI_CH_Config (U16 CardNumber, U16Channel, U16 AdRange_RefGnd)	Configuration of trigger source etc.
I16 D2K_AI_AsyncDbIBufferMode (U16 CardNumber, BOOLEAN Enable)	Configure of double buffer
I16 D2K_AI_ContBufferSetup (U16 CardNumber, void *Buffer, U32 ReadCount, U16 *BufferId)	Configuration of buffer for continuous analog signal
I16 D2K_AI_AsyncDbIBufferHalfReady (U16 CardNumber, BOOLEAN *HalfReady, BOOLEAN *StopFlag)	Judge the other buffer is ready or not, under the double buffer model
I16 D2K_AI_ContScanChannels (U16 CardNumber, U16 Channel, U16 BufId, U32 ReadScans, U32 ScanIntrv, U32 SampIntrv, U16 SyncMode)	Acquire signal under the setting acquisition frequency
I16 D2K_AI_AsyncClear (U16 CardNumber, U32 *StartPos, U32 *AccessCnt)	Stop the operation of asynchronous analog signal
I16 D2K_Release_Card (U16 CardNumber)	Release the card

3 AN ADAPTIVE CONFIGURATION METHOD OF MULTI-CARD

3.1 Design idea

According to the system demand, test cards will be configured adaptively by the method. After a certain experimental comparison, optimal configuration parameters of test cards can be found out. The basic steps are as follows:

1. At first, the type and quantity of the needed card should be determined according to the function and performance index of the system.

2. Assemble the cards and related hardware. Install the drivers and run the adaptive configuration programs. Screen the optimal configuration parameters

3. Configure the system according to the optimal configuration parameters.

All the heterogeneous board configuration parameters would be described into a unified form by XML[4]. And a unified API call for the operating system will be provided as well[5]. The method is mainly to complete the protocol adapting and data format conversion. The acquired data will be converted to unified data format after a pretreatment for the operating system. An adaptive item template will generate based on the various configuration parameters. Then, the automatic adaptation process will be done. The optimal configuration parameters of test cards can be found out and provided to the developers of the ground test system.

3.2 Realization of the method

1. Design of adaptive configuration function libraries

The adaptive configuration is composed of 4 function libraries. They are ConfigMain library, XMLConf library, DriverSetup library and SystemTest library.

The ConfigMain library is used to read the configuration parameters of all the cards, run the driver, run the test script and update the configuration parameters according to the test results.

The XMLConf library is used to open and close XML file, read and write parameters.

The DriverSetup library is used to install the drivers of the test cards.

The SystemTest library is used to test the system based on the test conditions, which mainly includes function, performance and stability.

The calling relationship of the 4 function libraries is shown in Fig.1.

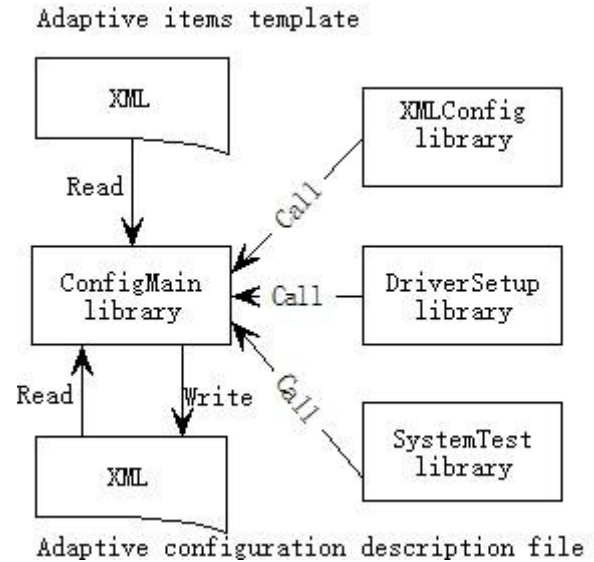


Fig 1 Calling relationship of 4 function libraries

2. Workflow of the adaptive configuration function

The workflow of the adaptive configuration function is shown in Fig.2.

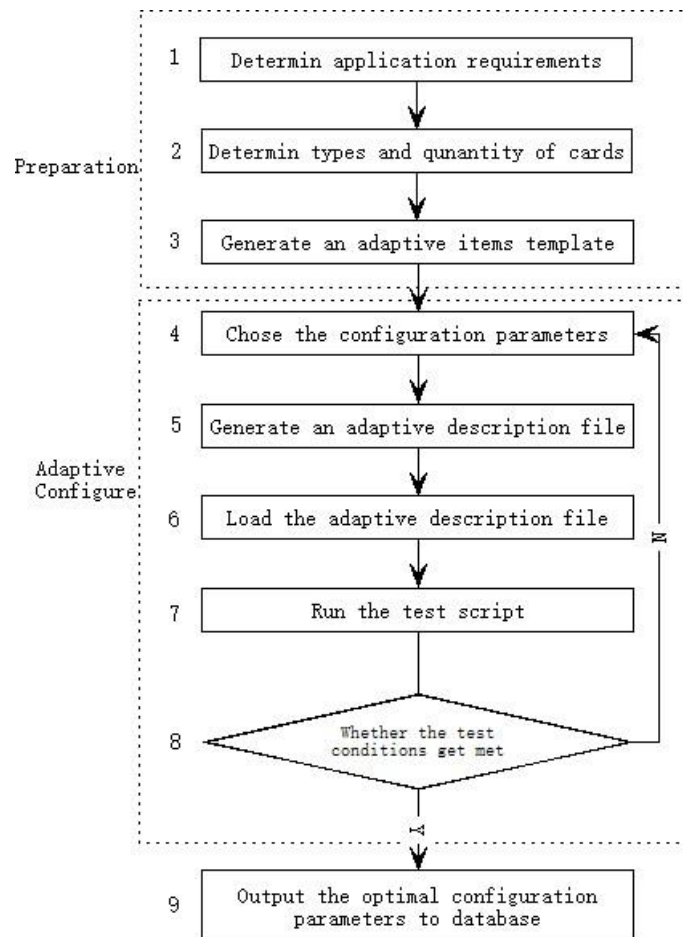


Fig 2 Workflow of the adaptive configuration function

The type of all the files is XML, which is easy and convenient to exchange the data. Here the workflow of the adaptive configuration is divided into two parts. One is the process of preparation, the

other is the process of adaptive configuration. The detailed workflow is explained as follows:

Step 1: Determine application requirements

The application requirements are decided by the types, quantity, sampling frequency and sampling precision of the signals.

Step 2: Determine types and quantity of cards

The types and quantity of the test cards can be determined by the application requirements which are determined in Step 1.

Step 3: Generate an adaptive items template

Due to each card has its own configuration file and the type and format are different, a unified adaptive items template should get generated. So a program to transform the configurations files is needed. The design idea of this program is shown in Fig.3.

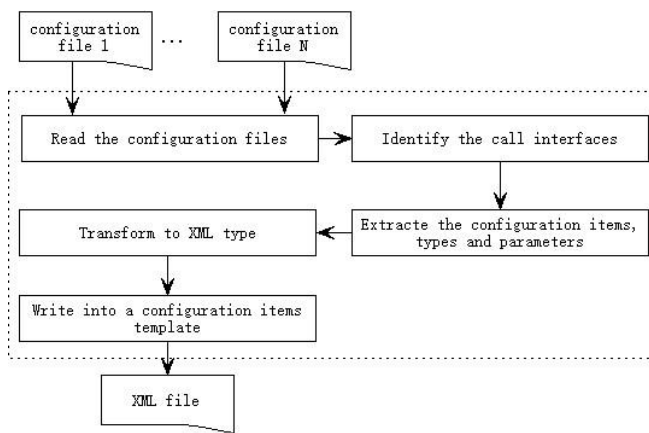


Fig 3 design idea of the adaptive configuration function

The adaptive configuration items template is XML files as well. Its structure is designed and shown in table 2.

Table 2. Structure adaptive configuration items template .

Index	items	description
1	name	Name of card
2	type	Type of card
3	version	Version of card
4	depend	Depended cards
5	priority	Priority of card
6	info	Description info
7	item	Parameters items
8	value	Value of item
9	weight	Weight of the last value

When a card needs to work, maybe some other cards are need at the same time. And when the priority of one card id higher than others, it should be get more consideration firstly. There are kinds of items for each card. And each item has many possible values. The weight of the value is the key part of adaptive configuration method. And the changes of weight decide the changes of value. So

the way of weight changes should be designed scientifically and reasonably. Each configuration item and the configuration parameters are corresponding to each other. That is each has the only optimal configuration parameter.

Step 4: Chose the configuration parameters

To each adaptive configuration item, the value with the largest weight should be chosen as the determined parameter.

Step 5: Generate an adaptive configuration description file

An adaptive description file should get generated based on the chosen parameters in Step 4. During the process of adaptive configuring, the adaptive configuration program reads, modifies and optimizes the parameters in the adaptive description file constantly. And the global optimal parameters would get out and get saved into the adaptive description file at last. For each configuration item, a new data structure is defined. The conceptual model of the data structure is shown as follows:

```

Data structure{
Card_ID,
Item_i,
Value_i1,...,Value_iN,
Weight_i1,...,Weight_iN
}
  
```

Card_ID is the ID of one test card. Item_i is one configuration item of the card. Value_i1, ... , Value_iN are all the possible values of the configuration item. Weight_i1, ... , Weight_iN are the weights to each value of the item.

Step 6: Load the adaptive configuration description file

The main adaptive configuration program will read the adaptive configuration description file generated in Step 5 and pick up the configuration items and configuration parameters based on the ID of each test card.

Step 7: Run the test script

The main adaptive configuration program will call the DriverSetup function. The drivers will get installed and the configuration parameters will get determined by the chosen configuration items and configuration parameters in Step 6. After other needed software and hardware get ready, the test script from script database will get executed to test the functions, performance and stability of the application system.

Step 8: Whether the test result met the test conditions

It is supposed that the number of all possible values for each configuration item is M. And the initial weight of each item is 1/M. No matter how the weights change, the sum of all weights is 1. The change of weight is the key to this adaptive configuration method. In this paper nonlinear genetic algorithm (NGA) is firstly introduced into the adaptive configuration[6]. Its main part is that the

factor of crossover and mutation probability was designed for adaptive changes according to the hormonal regulation of law, which reflects the main idea of this method.

1) Fitness function

Weight of each value is the chromosome. During each adaptive configuration process, the changes of weight are based on the operations of crossover and mutation. The fitness function is designed on the test conditions, which mainly includes function, performance and stability. Its formalization description is shown as follows.

$$Fitness = \omega_1 * (1 - a^{function}) + \omega_2 * (1 - a^{performance}) + \omega_3 * (1 - a^{stability}) \quad (1)$$

2) Crossover probability and mutation probability

The crossover probability and mutation probability of the nonlinear genetic algorithm are shown as follows.

$$P_c = P_c^0 (1 + \alpha \frac{(f_{av})^{nc}}{(f_{max} - f_{min})^{nc} + (f_{av})^{nc}}) \quad (2)$$

$$P_m = P_m^0 (1 + \beta \frac{(f_{av})^{nm}}{(f_{max} - f_{min})^{nm} + (f_{av})^{nm}}) \quad (3)$$

The detailed description of each parameter is explained in the paper[7].

Step 9: Output the optimal configuration parameters to database

If the adaptive configuration is successful, the last optimal configuration parameters would be saved into database. With the data increasing, the adaptive configuration would become more efficient.

Developers of launch vehicle ground test system can configure the system by the optimal configuration parameters and do some more optimization as well.

4 CONCLUSION

In the paper, an adaptive configuration method to multi-cards in ground test system is introduced in detail. This adaptive configuration method can solve the problem caused by multi-cards used in launch vehicle ground test system. The practice shows the efficiency and intelligence of the method.

REFERENCES

- [1] XIE Wei-qi. Research and implementation of digital detection technology for rockets and ground signal. Beijing: The academy of equipment, 2010.
- [2] LIU Zheng-gao. Overall plan to ensure aerospace software product. Quality and reliability, 2005, 6: 29-31.
- [3] ADLINK TECHNOLOGY INC.DAQ-/DAQe-/PXI-2204/2205/2206/2208 User's Manual.v.1.21, 2003.
- [4] Milo T, Suciu D, Vianu V. Typechecking for XML transformers. Journal of Computer and System Sciences, 2003, 66:66-97.
- [5] Xu Hong-bo, Xu Bao-guo, Xu Mai. Windows interface technology based on API. Journal of Shanghai University of Electric Power, 2001, (4): 43-46.
- [6] WANG Jun-hong, LIU Bao, YUAN Ruo-quan etc. Design and application of a nonlinear genetic algorithm. Computer simulation, 2010, 27(2):196-199.
- [7] XU Guo-feng, CAI Yuan-wen. Research on Learning and Evolutionary Algorithm of Agent for Task Oriented. Advanced Materials Research, 2014, 989-994:2033-2036.